

## Manipulations de fichiers

avril 2019

Remerciements infinis à Stéphane Gonnord.

### Débuter dans de bonnes conditions

Créer un répertoire de travail pour ce TP, et y récupérer les fichiers `communes.csv`, `premierspremiers.txt`, `matrix.txt`, `sudoku.txt` fournis à l'adresse

<https://diu-eil.gricad-pages.univ-grenoble-alpes.fr/donnees-programmation/>

Penser également à positionner le répertoire de travail de votre IDE dans ce même répertoire.

#### Exemple 1 — Pour ceux qui sont joueurs...

Il est possible par exemple de récupérer l'archive sans sortir de Python :

```
import urllib.request
urllib.request.urlretrieve(
    'https://diu-eil.gricad-pages.univ-grenoble-alpes.fr/'
    + 'donnees-programmation/TP-Fichiers/materiel-tp-fichiers.tar',
    './materiel-tp-fichiers.tar')
```

Pour ceux qui sont très joueurs : chercher sur <https://docs.python.org/3/library/> comment déballer l'archive dans Python également.

## 1 Boîte à outils

### 1.1 Manipulations de base

| Quoi   | Comment  |
|--|--|
| Ouverture en lecture                                       | <code>monfichier = open('lenom.txt', 'r')</code>                                 |
| Ouverture en écriture                                      | <code>monfichier = open('lenom.txt', 'w')</code>                                 |
| Fermeture  | <code>monfichier.close()</code>  |
| Lire la ligne suivante                                     | <code>ligne = monfichier.readline()</code>                                       |
| Parcourir toutes les lignes                                | <code>for ligne in monfichier.readlines():</code><br>...                         |
| Écrire dans le fichier...<br>et passer à la ligne suivante | <code>monfichier.write('plof')</code><br><code>monfichier.write('plof\n')</code> |

Comme d'habitude, <https://docs.python.org> pour aller plus loin.

### Exemple 2 — Écrire les 1000 premiers premiers dans un fichier

```
def mille_premiers():
    fileout = open('premierspremiers.txt','w')

    nb,k = 0,2

    while nb < 1000:
        if premier(k):
            fileout.write(str(k)+'\n')
            nb += 1
            k += 1

    fileout.close()
```

(Une fonction `premier` est fournie dans les annexes.)

### Exemple 3 — Et pour les lire en les sommant

```
def somme_premiers():
    filein = open('premierspremiers.txt','r')
    s = 0
    for l in filein.readlines():
        s += int(l)
    filein.close()
```

## 1.2 Split et strip

| Quoi  | Comment   |
|---|---|
| Séparer une chaîne selon les symboles ':'<br>ou selon les tabulations | <code>toto.split(':')</code><br><code>toto.split('\t')</code> |
| Supprimer les blancs en début et en fin de ligne                      | <code>toto.strip()</code>                                     |

Pour aller plus loin : `help(str)`

### Exemple 4 — Par blocs de 10

Imaginons les 100 premiers nombres premiers écrits dans un fichier, mais regroupés par 10 (c'est l'objet de l'exercice 1), et séparés par des tabulations. Pour les lire et les mettre dans une liste, ça pourrait donner :

```
def lecture_par_10():
    filein = open('premiers-par-10.txt','r')
    lp = []
    for l in filein.readlines():
        for i in l.strip().split('\t'):
            lp.append(int(i))
    filein.close()
```

## 2 Exercices sur des fichiers texte

### 2.1 Écriture des nombres premiers par 10

#### EXERCICE 1

Écrire une fonction qui lit les 1000 premiers nombres premiers dans le fichier `premierpremier.txt`... et les place dans une liste. Écrire ensuite dans un nouveau fichier ces 1000 entiers regroupés par 10 et séparés par des tabulations. Ouvrir le fichier créé avec votre éditeur préféré.

#### EXERCICE 2 (POUR OCCUPER LES PLUS RAPIDES)

Calculer tous les nombres premiers inférieurs à  $10^6$ . Ces nombres seront mis par lignes de dix (séparés par des virgules) dans dix fichiers traitant des tranches de longueur  $10^5$ .

Par exemple, la première ligne de mon fichier `Tranche8.txt` est

800011,800029,800053,800057,800077,800083,800089,800113,800117,800119

et la dernière est<sup>1</sup> :

899939,899971,899981,

Combien d'entiers inférieurs à  $10^6$  sont premiers ?

### 2.2 Un peu plus dur

#### EXERCICE 3

Un fichier texte contient 35 442 lignes de la forme :

| Nom de la région     | Code département | Nom de la commune        | Population totale |
|----------------------|------------------|--------------------------|-------------------|
| Auvergne-Rhône-Alpes | 01               | L' Abergement-Clémenciat | 785               |
| Auvergne-Rhône-Alpes | 01               | L' Abergement-de-Varey   | 242               |
| Auvergne-Rhône-Alpes | 01               | Ambérieu-en-Bugey        | 14631             |
| Auvergne-Rhône-Alpes | 01               | Ambérieux-en-Dombes      | 1652              |
| Auvergne-Rhône-Alpes | 01               | Ambléon                  | 115               |

(Des tabulations séparent les quatre champs ; même si un nom comporte des espaces il ne constitue qu'un seul champ.)

Écrire une fonction qui lit ce fichier, détermine le nombre total de communes de plus de 10000 habitants, et le nombre de communes dans chaque département.

Données INSEE : <https://insee.fr/fr/statistiques/3545833?sommaire=3292701>

#### EXERCICE 4 (LECTURE DANS UNE MATRICE)

Écrire une fonction qui lit dans une matrice (80, 80) les entiers présents dans le fichier `matrix.txt`.

#### EXERCICE 5 (PROJECT EULER #81)

Dans la matrice  $5 \times 5$  ci-dessous, le chemin de somme minimale d'en haut à gauche jusqu'en bas à droite, obtenu **uniquement par des déplacements vers la droite et vers le bas**, est indiqué en gras et vaut 2427.

$$\begin{pmatrix} \mathbf{131} & 673 & 234 & 103 & 18 \\ \mathbf{201} & \mathbf{96} & \mathbf{342} & 965 & 150 \\ 630 & 803 & \mathbf{746} & \mathbf{422} & 111 \\ 537 & 699 & 497 & \mathbf{121} & 956 \\ 805 & 732 & 524 & \mathbf{37} & \mathbf{331} \end{pmatrix}$$

Trouver le chemin de somme minimale dans `matrix.txt`, toujours d'en haut à gauche jusqu'en bas à droite.

**Indications :** Si on note  $\varphi(i, j)$  la somme minimale pour aller jusqu'à la case  $(i, j)$ , les cas de bord sont simples, et puisque « on vient d'en haut ou de la gauche », on a une relation simple permettant de calculer les  $\varphi(i, j)$  par programmation dynamique/mémoïsation. On doit trouver 427337.

Si vous trouvez 428433, regardez comment vous avez créé la matrice destinée à tabuler les  $\varphi(i, j)$ ...

---

1. Et oui, je sais que ce serait mieux sans la virgule finale !

## 2.3 Des grilles de sudoku

### EXERCICE 6

Dans le fichier `sudoku.txt`, on trouve 50 grilles de sudoku. Ouvrez ce fichier avec un éditeur, histoire de voir sa tête. Importez les 50 grilles pour en faire une liste de 50 listes de 9 listes de 9 entiers !

Les deux exercices suivants sont très optionnels...

### EXERCICE 7

Affichez les grilles en mode texte de façon plus lisible. Par exemple :

```
>>> print(prettyprinting(grids[10]))
```

```
-----  
|   | 1 2 5|4   |  
|   8|4   |   |  
|4 2 |8   |   |  
-----  
| 3 |   | 9 5|  
| 6 |9  2| 1 |  
|5 1 |   | 6  |  
-----  
|   |   3| 4 9|  
|   |   7|2   |  
|  1|2 9 8|   |  
-----
```

On aura noté que la fonction `prettyprinting` renvoie une chaîne, plutôt que de réaliser elle-même l'affichage via `print` sans rien renvoyer.

### Bonus : créer des pdf simplement

Une façon basique de procéder : dessiner des choses dans un *canevas*.

| Quoi                         | Comment  |
|------------------------------|--|
| La librairie en jeu          | <code>from reportlab.pdfgen import canvas</code>   |
| Créer un canevas             | <code>Monpdf = canvas.Canvas('tangents.pdf')</code>  |
| Mettre des choses dedans     | <code>Monpdf.drawString(100,300,'Hello world')</code><br><code>Monpdf.line(x0,y0,x1,y1)</code><br><code>Monpdf.rect(x0,y0,deltax,deltay)</code><br><code>Monpdf.circle(400,400,100,fill=True)</code> |
| Changer des caractéristiques | <code>Monpdf.setFont("Helvetica",14)</code><br><code>Monpdf.setStrokeColorRGB(255,0,0)</code><br><code>Monpdf.setFillColorRGB(255,255,255)</code>  |
| Le sauver                    | <code>Monpdf.save()</code>   |

### EXERCICE 8

Vous avez trouvé le résultat de l'exercice précédent vraiment laid ? C'est normal. Il convient donc de créer 50 jolis pdf... Un exemple de rendu possible est donné en annexe.

### 3 Annexes

#### 3.1 Primalité

##### Exemple 5 — Un code basique

```
def premier(n):  
    if n == 2 or n == 3: return True  
    if n == 1 or n%2==0: return False  
    k = 3  
    while k**2 <= n:  
        if n % k == 0: return False  
        k += 2  
    return True
```

#### 3.2 Pour occuper les cancre

Grille numéro 11

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
|   |   |   | 1 | 2 | 5 | 4 |   |   |
|   |   | 8 | 4 |   |   |   |   |   |
| 4 | 2 |   | 8 |   |   |   |   |   |
|   | 3 |   |   |   |   |   | 9 | 5 |
|   | 6 |   | 9 |   | 2 |   | 1 |   |
| 5 | 1 |   |   |   |   |   | 6 |   |
|   |   |   |   |   | 3 |   | 4 | 9 |
|   |   |   |   |   | 7 | 2 |   |   |
|   |   | 1 | 2 | 9 | 8 |   |   |   |